30 | 18 |

1 30 18

1	30	18
	18	

1	30	18
	18	
	12	

1	30	18	1
	18		
	12		

1	30	18	1
	18	12	
	12		

1	30	18	1
	18	12	
	12	6	

1	30	18	1
	18	12	
2	12	6	

1	30	18	1
	18	12	
2	12	6	
	12		

1	30	18	1
	18	12	
2	12	6	
	12		
	0		

```
0012FF78
                                                       0012FF79
if( divisor > dividend )
                                       di vi sor
                                                  30
                                                       0012FF7A
                                                       0012FF7B
   di vi sor = di vi dend;
   di vi dend = di vi sor;
                                                       0012FF7C
}
                                                       0012FF7D
                                      di vi dend
                                                  18
                                                       0012FF7E
                                                       0012FF7F
```

```
0012FF78
                                                       0012FF79
if( divisor > dividend )
                                       di vi sor
                                                  18
                                                       0012FF7A
                                                       0012FF7B
   di vi sor = di vi dend;
   di vi dend = di vi sor;
                                                       0012FF7C
}
                                                       0012FF7D
                                      di vi dend
                                                  18
                                                       0012FF7E
                                                       0012FF7F
```

```
0012FF74
                                                      0012FF75
                                       buffer
                                                      0012FF76
if( divisor > dividend )
                                                      0012FF77
                                                      0012FF78
   buffer = divisor;
   di vi sor = di vi dend;
                                                      0012FF79
                                      di vi sor
                                                 30
   di vi dend = buffer;
                                                      0012FF7A
                                                      0012FF7B
                                                      0012FF7C
                                                      0012FF7D
                                     di vi dend
                                                 18
                                                      0012FF7E
                                                      0012FF7F
```

```
0012FF74
                                                      0012FF75
                                       buffer
                                                 30
                                                      0012FF76
if( divisor > dividend )
                                                      0012FF77
                                                      0012FF78
   buffer = divisor;
   di vi sor = di vi dend;
                                                      0012FF79
                                      di vi sor
                                                 30
   di vi dend = buffer;
                                                      0012FF7A
                                                      0012FF7B
                                                      0012FF7C
                                                      0012FF7D
                                     di vi dend
                                                 18
                                                      0012FF7E
                                                      0012FF7F
```

```
0012FF74
                                                      0012FF75
                                       buffer
                                                 30
                                                      0012FF76
if( divisor > dividend )
                                                      0012FF77
                                                      0012FF78
   buffer = divisor;
   di vi sor = di vi dend;
                                                      0012FF79
                                      di vi sor
                                                 18
   di vi dend = buffer;
                                                      0012FF7A
                                                      0012FF7B
                                                      0012FF7C
                                                      0012FF7D
                                     di vi dend
                                                 18
                                                      0012FF7E
                                                      0012FF7F
```

```
0012FF74
                                                      0012FF75
                                       buffer
                                                 30
                                                      0012FF76
if( divisor > dividend )
                                                      0012FF77
                                                      0012FF78
   buffer = divisor;
   di vi sor = di vi dend;
                                                      0012FF79
                                      di vi sor
                                                 18
   di vi dend = buffer;
                                                      0012FF7A
                                                      0012FF7B
                                                      0012FF7C
                                                      0012FF7D
                                     di vi dend
                                                 30
                                                      0012FF7E
                                                      0012FF7F
```

1	30	18	1
	18	12	
2	12	6	
	12		
	0		

di vi sor	18	00033AE8
di vi dend	30	00033AEC
buf[0]		00033AF0
buf[1]		00033AF4
buf[2]		00033AF8
buf[3]		00033AFC
buf[4]		00033B00
buf[5]		00033B04
buf[6]		00033B08
•		•
Į.		

1	30	18	1
	18	12	
2	12	6	
	12		
	0		

di vi sor	18	00033AE8
di vi dend	30	00033AEC
buf[0]	30	00033AF0
buf[1]		00033AF4
buf[2]		00033AF8
buf[3]		00033AFC
buf[4]		00033В00
buf[5]		00033В04
buf[6]		00033В08
•		:
		ı

1	30	18	1
	18	12	
2	12	6	
	12		
	0		

di vi sor	18	00033AE8
di vi dend	30	00033AEC
buf[0]	30	00033AF0
buf[1]	18	00033AF4
buf[2]		00033AF8
buf[3]		00033AFC
buf[4]		00033B00
buf[5]		00033B04
buf[6]		00033B08
•		•

1	30	18	1
	18	12	
2	12	6	
	12		
	0		

di vi sor	18	00033AE8
di vi dend	30	00033AEC
buf[0]	30	00033AF0
buf[1]	18	00033AF4
buf[2]	12	00033AF8
buf[3]		00033AFC
buf[4]		00033B00
buf[5]		00033B04
buf[6]		00033B08
•		•

1	30	18	1
	18	12	
2	12	6	
	12		
	0		

di vi sor	18	00033AE8
di vi dend	30	00033AEC
buf[0]	30	00033AF0
buf[1]	18	00033AF4
buf[2]	12	00033AF8
buf[3]	6	00033AFC
buf[4]		00033B00
buf[5]		00033B04
buf[6]		00033B08
•		•

1	30	18	1
	18	12	
2	12	6	
	12		
	0		

di vi sor	18	00033AE8
di vi dend	30	00033AEC
buf[0]	30	00033AF0
buf[1]	18	00033AF4
buf[2]	12	00033AF8
buf[3]	6	00033AFC
buf[4]	0	00033B00
buf[5]		00033B04
buf[6]		00033B08
•		:
	-	=

```
buf[ 0 ] = di vi dend;
buf[ 1 ] = di vi sor;
buf[ 2 ] = buf[ 0 ] % buf[ 1 ];
buf[ 3 ] = buf[ 1 ] % buf[ 2 ];
buf[ 4 ] = buf[ 2 ] % buf[ 3 ];
cout << buf[ 3 ] << endl;</pre>
```

di vi sor	18	00033AE8
di vi dend	30	00033AEC
buf[0]		00033AF0
buf[1]		00033AF4
buf[2]		00033AF8
buf[3]		00033AFC
buf[4]		00033B00
buf[5]		00033B04
buf[6]		00033B08
•		•

```
buf[ 0 ] = di vi dend;
buf[ 1 ] = di vi sor;
buf[ 2 ] = buf[ 0 ] % buf[ 1 ];
buf[ 3 ] = buf[ 1 ] % buf[ 2 ];
buf[ 4 ] = buf[ 2 ] % buf[ 3 ];
cout << buf[ 3 ] << endl;</pre>
```

di vi sor	18	00033AE8
di vi dend	30	00033AEC
buf[0]	30	00033AF0
buf[1]	18	00033AF4
buf[2]		00033AF8
buf[3]		00033AFC
buf[4]		00033B00
buf[5]		00033B04
buf[6]		00033B08
•		:

```
buf[ 0 ] = di vi dend;
buf[ 1 ] = di vi sor;
buf[ 2 ] = buf[ 0 ] % buf[ 1 ];
buf[ 3 ] = buf[ 1 ] % buf[ 2 ];
buf[ 4 ] = buf[ 2 ] % buf[ 3 ];
cout << buf[ 3 ] << endl;</pre>
```

di vi sor	18	00033AE8
di vi dend	30	00033AEC
buf[0]	30	00033AF0
buf[1]	18	00033AF4
buf[2]	12	00033AF8
buf[3]		00033AFC
buf[4]		00033B00
buf[5]		00033B04
buf[6]		00033B08
•		:

```
buf[ 0 ] = di vi dend;
buf[ 1 ] = di vi sor;
buf[ 2 ] = buf[ 0 ] % buf[ 1 ];
buf[ 3 ] = buf[ 1 ] % buf[ 2 ];
buf[ 4 ] = buf[ 2 ] % buf[ 3 ];
cout << buf[ 3 ] << endl;</pre>
```

di vi sor	18	00033AE8
di vi dend	30	00033AEC
buf[0]	30	00033AF0
buf[1]	18	00033AF4
buf[2]	12	00033AF8
buf[3]	6	00033AFC
buf[4]		00033B00
buf[5]		00033B04
buf[6]		00033B08
•		:

```
buf[ 0 ] = di vi dend;
buf[ 1 ] = di vi sor;
buf[ 2 ] = buf[ 0 ] % buf[ 1 ];
buf[ 3 ] = buf[ 1 ] % buf[ 2 ];
buf[ 4 ] = buf[ 2 ] % buf[ 3 ];
cout << buf[ 3 ] << endl;</pre>
```

di vi sor	18	00033AE8
di vi dend	30	00033AEC
buf[0]	30	00033AF0
buf[1]	18	00033AF4
buf[2]	12	00033AF8
buf[3]	6	00033AFC
buf[4]	0	00033B00
buf[5]		00033B04
buf[6]		00033B08
:		:

```
buf[0] = dividend;
buf[ 1 ] = di vi sor;
buf[ 2 ] = buf[ 0 ] % buf[ 1 ];
buf[ 3 ] = buf[ 1 ] % buf[ 2 ];
buf[ 4 ] = buf[ 2 ] % buf[ 3 ];
cout << buf[ 3 ] << endl;
buf[0] = dividend;
buf[ 1 ] = di vi sor;
for( int i{ 0 }; != 0; i++ )
  buf[i + 2] = buf[i] \% buf[i + 1];
cout << buf[ ] << endl << endl;</pre>
```

```
buf[0] = dividend;
buf[ 1 ] = di vi sor;
buf[ 2 ] = buf[ 0 ] % buf[ 1 ];
buf[ 3 ] = buf[ 1 ] % buf[ 2 ];
buf[ 4 ] = buf[ 2 ] % buf[ 3 ];
cout << buf[ 3 ] << endl;
buf[0] = dividend;
buf[ 1 ] = di vi sor;
for( int i { 0 }; buf[ ] != 0; i++ )
  buf[i + 2] = buf[i] \% buf[i + 1];
cout << buf[ ] << endl << endl;</pre>
```

```
00033AE4
        i
 di vi sor
            18
                 00033AE8
di vi dend
                 00033AEC
            30
  buf[0]
                 00033AF0
  buf[1]
                 00033AF4
  buf [2]
                 00033AF8
  buf[3]
                 00033AFC
  buf [4]
                 00033B00
  buf[5]
                 00033B04
  buf [6]
                 00033B08
```

```
00033AE4
        i
 di vi sor
            18
                 00033AE8
di vi dend
                 00033AEC
            30
  buf[0]
                 00033AF0
            30
  buf[1]
            18
                 00033AF4
  buf[2]
                 00033AF8
  buf[3]
                 00033AFC
  buf [4]
                 00033B00
  buf[5]
                 00033B04
  buf[6]
                 00033B08
```

```
00033AE4
        i
 di vi sor
                 00033AE8
            18
di vi dend
                 00033AEC
            30
  buf[0]
                 00033AF0
            30
  buf[1]
            18
                 00033AF4
  buf [2]
                 00033AF8
  buf[3]
                 00033AFC
  buf [4]
                 00033B00
  buf[5]
                 00033B04
  buf[6]
                 00033B08
```

```
00033AE4
       i
 di vi sor
                 00033AE8
            18
                 00033AEC
di vi dend
            30
  buf[0]
                 00033AF0
            30
  buf[1]
            18
                 00033AF4
  buf[2]
                 00033AF8
            12
  buf[3]
                 00033AFC
  buf [4]
                 00033B00
  buf[5]
                 00033B04
  buf[6]
                 00033B08
```

```
00033AE4
        i
 di vi sor
                 00033AE8
            18
di vi dend
                 00033AEC
            30
  buf[0]
                 00033AF0
            30
  buf[1]
            18
                 00033AF4
  buf[2]
                 00033AF8
            12
  buf[3]
                 00033AFC
  buf [4]
                 00033B00
  buf[5]
                 00033B04
  buf[6]
                 00033B08
```

```
00033AE4
        i
 di vi sor
                 00033AE8
            18
di vi dend
                 00033AEC
            30
  buf[0]
                 00033AF0
            30
  buf[1]
            18
                 00033AF4
  buf[2]
                 00033AF8
            12
  buf[3]
                 00033AFC
            6
  buf [4]
                 00033B00
  buf[5]
                 00033B04
  buf[6]
                 00033B08
```

```
00033AE4
       i
 di vi sor
                 00033AE8
            18
di vi dend
                 00033AEC
            30
  buf[0]
                 00033AF0
            30
  buf[1]
            18
                 00033AF4
  buf[2]
                 00033AF8
            12
  buf[3]
                 00033AFC
            6
  buf [4]
                 00033B00
  buf[5]
                 00033B04
  buf[6]
                 00033B08
```

```
00033AE4
       i
 di vi sor
                 00033AE8
            18
di vi dend
                 00033AEC
            30
  buf[0]
                 00033AF0
            30
  buf[1]
            18
                 00033AF4
  buf[2]
                 00033AF8
            12
  buf[3]
                 00033AFC
            6
  buf [4]
                 00033B00
            0
  buf[5]
                 00033B04
  buf[6]
                 00033B08
```

```
00033AE4
       i
 di vi sor
                 00033AE8
            18
di vi dend
                 00033AEC
            30
  buf[0]
                 00033AF0
            30
  buf[1]
            18
                 00033AF4
  buf[2]
                 00033AF8
            12
  buf[3]
                 00033AFC
            6
  buf [4]
                 00033B00
            0
  buf[5]
                 00033B04
  buf[6]
                 00033B08
```

```
00033AE4
        i
 di vi sor
            18
                 00033AE8
di vi dend
                 00033AEC
            30
  buf[0]
                 00033AF0
  buf[1]
                 00033AF4
  buf [2]
                 00033AF8
  buf[3]
                 00033AFC
  buf [4]
                 00033B00
  buf[5]
                 00033B04
  buf [6]
                 00033B08
```

```
buf[ 0 ] = di vi dend;
buf[ 1 ] = di vi sor;
for( int i{ 0 }; buf[ i + 1 ] != 0; i++ )
  buf[ i + 2 ] = buf[ i ] % buf[ i + 1 ];
cout << buf[ i ] << endl << endl;</pre>
```

```
00033AE4
        i
 di vi sor
            18
                 00033AE8
di vi dend
                 00033AEC
            30
  buf[0]
                 00033AF0
            30
  buf[1]
            18
                 00033AF4
  buf[2]
                 00033AF8
  buf[3]
                 00033AFC
  buf [4]
                 00033B00
  buf[5]
                 00033B04
  buf[6]
                 00033B08
```

```
buf[ 0 ] = di vi dend;
buf[ 1 ] = di vi sor;
for( int i{ 0 }; buf[ i + 1 ] != 0; i++ )
  buf[ i + 2 ] = buf[ i ] % buf[ i + 1 ];
cout << buf[ i ] << endl << endl;</pre>
```

```
00033AE4
       i
 di vi sor
                 00033AE8
            18
di vi dend
                 00033AEC
            30
  buf[0]
                 00033AF0
            30
  buf[1]
            18
                 00033AF4
  buf[2]
                 00033AF8
  buf[3]
                 00033AFC
  buf [4]
                 00033B00
  buf[5]
                 00033B04
  buf[6]
                 00033B08
```

```
buf[ 0 ] = di vi dend;
buf[ 1 ] = di vi sor;
for( int i{ 0 }; buf[ i + 1 ] != 0; i++ )
  buf[ i + 2 ] = buf[ i ] % buf[ i + 1 ];
cout << buf[ i ] << endl << endl;</pre>
```

```
00033AE4
       i
 di vi sor
                 00033AE8
            18
di vi dend
                 00033AEC
            30
  buf[0]
                 00033AF0
            30
  buf[1]
            18
                 00033AF4
  buf[2]
                 00033AF8
            12
  buf[3]
                 00033AFC
  buf [4]
                 00033B00
  buf[5]
                 00033B04
  buf[6]
                 00033B08
```

```
buf[ 0 ] = di vi dend;
buf[ 1 ] = di vi sor;
for( int i{ 0 }; buf[ i + 1 ] != 0; i++ )
   buf[ i + 2 ] = buf[ i ] % buf[ i + 1 ];
cout << buf[ i ] << endl << endl;</pre>
```

```
00033AE4
        i
 di vi sor
                 00033AE8
            18
di vi dend
                 00033AEC
            30
  buf[0]
                 00033AF0
            30
  buf[1]
            18
                 00033AF4
  buf[2]
                 00033AF8
            12
  buf[3]
                 00033AFC
  buf [4]
                 00033B00
  buf[5]
                 00033B04
  buf[6]
                 00033B08
```

```
buf[ 0 ] = di vi dend;
buf[ 1 ] = di vi sor;
for( int i{ 0 }; buf[ i + 1 ] != 0; i++ )
   buf[ i + 2 ] = buf[ i ] % buf[ i + 1 ];
cout << buf[ i ] << endl << endl;</pre>
```

```
00033AE4
        i
 di vi sor
                 00033AE8
            18
di vi dend
                 00033AEC
            30
  buf[0]
                 00033AF0
            30
  buf[1]
            18
                 00033AF4
  buf[2]
                 00033AF8
            12
  buf[3]
                 00033AFC
            6
  buf [4]
                 00033B00
  buf[5]
                 00033B04
  buf[6]
                 00033B08
```

```
buf[ 0 ] = di vi dend;
buf[ 1 ] = di vi sor;
for( int i{ 0 }; buf[ i + 1 ] != 0; i++ )
  buf[ i + 2 ] = buf[ i ] % buf[ i + 1 ];
cout << buf[ i ] << endl << endl;</pre>
```

```
00033AE4
       i
 di vi sor
                 00033AE8
            18
di vi dend
                 00033AEC
            30
  buf[0]
                 00033AF0
            30
  buf[1]
            18
                 00033AF4
  buf[2]
                 00033AF8
            12
  buf[3]
                 00033AFC
            6
  buf [4]
                 00033B00
  buf[5]
                 00033B04
  buf[6]
                 00033B08
```

```
buf[ 0 ] = di vi dend;
buf[ 1 ] = di vi sor;
for( int i{ 0 }; buf[ i + 1 ] != 0; i++ )
  buf[ i + 2 ] = buf[ i ] % buf[ i + 1 ];
cout << buf[ i ] << endl << endl;</pre>
```

```
00033AE4
       i
 di vi sor
                 00033AE8
            18
di vi dend
                 00033AEC
            30
  buf[0]
                 00033AF0
            30
  buf[1]
            18
                 00033AF4
  buf[2]
                 00033AF8
            12
  buf[3]
                 00033AFC
            6
  buf [4]
                 00033B00
            0
  buf[5]
                 00033B04
  buf[6]
                 00033B08
```

```
buf[ 0 ] = di vi dend;
buf[ 1 ] = di vi sor;
for( int i{ 0 }; buf[ i + 1 ] != 0; i++ )
  buf[ i + 2 ] = buf[ i ] % buf[ i + 1 ];
cout << buf[ i ] << endl << endl;</pre>
```

```
00033AE4
       i
 di vi sor
                 00033AE8
            18
di vi dend
                 00033AEC
            30
  buf[0]
                 00033AF0
            30
  buf[1]
            18
                 00033AF4
  buf[2]
                 00033AF8
            12
  buf[3]
                 00033AFC
            6
  buf [4]
                 00033B00
            0
  buf[5]
                 00033B04
  buf[6]
                 00033B08
```

```
buf[ 0 ] = di vi dend;
buf[ 1 ] = di vi sor;
for( int i{ 0 }; buf[ i + 1 ] != 0; i++ )
   buf[ i + 2 ] = buf[ i ] % buf[ i + 1 ];
cout << buf[ i ] << endl << endl;</pre>
```

```
int main()
{
   int dividend, divisor;
   int hold;
   cin >> di vi dend >> di vi sor;
   if( divisor > dividend )
      hold = divisor:
      di vi sor = di vi dend;
      di vi dend = hol d;
   int buffer[ 258340 ];
   buffer[ 0 ] = di vi dend;
   buffer[ 1 ] = di vi sor;
   for( int i{ 0 }; buffer[ i + 1 ] != 0; i++ )
      buffer[ i + 2 ] = buffer[ i ] % buffer[ i + 1 ];
   cout << buffer[ i ] << endl << endl;</pre>
}
```

Don't use arrays

0012F				
remainder 0012F				
0010	18	$0 \mid$	$\frac{1}{30}$	1
0012F				•
2 0012F	12	8	18	
di vi sor 18 0012F	6	2	12	2
0012F				_
0012F		2	12	
0012F		$\overline{0}$		
di vi dend 30 0012F	l	٠		
dividend 30 0012F				
0012F				

		0012FF74
ı ı remai nder	12	0012FF75
30 18 1	12	0012FF76
		0012FF77
18 12		0012FF78
12 6 di vi sor	18	0012FF79
		0012FF7A
12		0012FF7B
		0012FF7C
di vi dend	30	0012FF7D
di vi dene		0012FF7E
		0012FF7F

				remai nder		0012FF74
					12	0012FF75
1	30	18	1	i ellai ildei	12	0012FF76
			-			0012FF77
	18	12				0012FF78
2	12	6		di vi sor	18	0012FF79
		O		ui vi soi	10	0012FF7A
	12					0012FF7B
	0					0012FF7C
l	٠			di vi dend	18	0012FF7D
				ui vi uenu	10	0012FF7E
						0012FF7F

		0012FF74
remai nder	12	0012FF75
1 30 18 1	12	0012FF76
		0012FF77
18 12		0012FF78
2 12 6 di vi sor	12	0012FF79
di vi soi	12	0012FF7A
12		0012FF7B
		0012FF7C
	18	0012FF7D
di vi delid		0012FF7E
		0012FF7F

0012FF74					
6 0012FF75	6	remai nder		_	
0012FF76		1 emarrider	18	30	1
0012FF77					_
0012FF78			12	18	
0012FF79	12	di vi sor	6	12	2
0012FF7A	12	ui vi soi			_
0012FF7B				12	
0012FF7C				0	
0012FF7D	18	di vi dend			
0012FF7E		ui vi dend			
0012FF7F					

0012FF74					
0012FF75	6	remai nder			
0012FF76		1 emarrider	18	30	1
0012FF77		•			•
0012FF78			12	18	
0012FF79	12	di vi sor	6	12	2
0012FF7A	12	ui vi soi			_
0012FF7B				12	
0012FF7C				0	
0012FF7D	12	di vi dend			
0012FF7E	12	ui vi uenu			
0012FF7F					

0012FF74						
6 0012FF75	6	remai nder		_		
0012FF76	0	1 ellar nder	18	1	30	1
0012FF77						_
0012FF78			12	1	18	
0012FF79	6	di vi sor	6		12	2
0012FF7A		ui vi soi				_
0012FF7B					12	
0012FF7C	12				0	
0012FF7D		di vi dend	l			
0012FF7E		ui vi dend				
0012FF7F						

	remai nder				
0					
	1 emarinder	$\mid \mid $	18	30	1
					_
			12	18	
6	di vi sor		6	12	2
	ui vi soi				_
				12	
				0	
12	di vi dend	l	1		
12	ui vi dend				

```
0012FF74
                                                              0012FF75
                                            remai nder
                                                              0012FF76
remainder = dividend % divisor;
                                                              0012FF77
while( remainder != 0 )
                                                              0012FF78
                                                              0012FF79
  di vi dend = di vi sor;
                                               di vi sor
                                                         18
                                                              0012FF7A
  divisor = remainder:
                                                              0012FF7B
  remainder = dividend % divisor;
                                                              0012FF7C
                                                              0012FF7D
                                             di vi dend
                                                         30
                                                              0012FF7E
                                                              0012FF7F
```

```
0012FF74
                                                              0012FF75
                                            remai nder
                                                         12
                                                              0012FF76
remainder = dividend % divisor;
                                                              0012FF77
while( remainder != 0 )
                                                              0012FF78
                                                              0012FF79
  di vi dend = di vi sor;
                                               di vi sor
                                                         18
                                                              0012FF7A
  divisor = remainder:
                                                              0012FF7B
  remainder = dividend % divisor;
                                                              0012FF7C
                                                              0012FF7D
                                              di vi dend
                                                         30
                                                              0012FF7E
                                                              0012FF7F
```

```
0012FF74
                                                               0012FF75
                                             remai nder
                                                          12
                                                               0012FF76
remainder = dividend % divisor;
                                                               0012FF77
while( remainder != 0 )
                                                               0012FF78
                                                               0012FF79
  di vi dend = di vi sor;
                                               di vi sor
                                                          18
                                                               0012FF7A
  divisor = remainder:
                                                               0012FF7B
  remainder = dividend % divisor;
}
                                                               0012FF7C
                                                               0012FF7D
                                              di vi dend
                                                          18
                                                               0012FF7E
                                                               0012FF7F
```

```
0012FF74
                                                               0012FF75
                                             remai nder
                                                          12
                                                               0012FF76
remainder = dividend % divisor;
                                                               0012FF77
while( remainder != 0 )
                                                               0012FF78
                                                               0012FF79
  di vi dend = di vi sor;
                                               di vi sor
                                                          12
                                                               0012FF7A
  divisor = remainder:
                                                               0012FF7B
  remainder = dividend % divisor;
}
                                                               0012FF7C
                                                               0012FF7D
                                              di vi dend
                                                          18
                                                               0012FF7E
                                                               0012FF7F
```

```
0012FF74
                                                              0012FF75
                                            remai nder
                                                              0012FF76
remainder = dividend % divisor;
                                                              0012FF77
while( remainder != 0 )
                                                              0012FF78
                                                              0012FF79
  di vi dend = di vi sor;
                                               di vi sor
                                                         12
                                                              0012FF7A
  divisor = remainder:
                                                              0012FF7B
  remainder = dividend % divisor;
                                                              0012FF7C
                                                              0012FF7D
                                             di vi dend
                                                         18
                                                              0012FF7E
                                                              0012FF7F
```

```
0012FF74
                                                              0012FF75
                                            remai nder
                                                              0012FF76
remainder = dividend % divisor;
                                                              0012FF77
while( remainder != 0 )
                                                              0012FF78
                                                              0012FF79
  di vi dend = di vi sor;
                                               di vi sor
                                                         12
                                                              0012FF7A
  divisor = remainder:
                                                              0012FF7B
  remainder = dividend % divisor;
                                                              0012FF7C
                                                              0012FF7D
                                             di vi dend
                                                         12
                                                              0012FF7E
                                                              0012FF7F
```

```
0012FF74
                                                              0012FF75
                                            remai nder
                                                              0012FF76
remainder = dividend % divisor;
                                                              0012FF77
while( remainder != 0 )
                                                              0012FF78
                                                              0012FF79
  di vi dend = di vi sor;
                                               di vi sor
                                                              0012FF7A
  divisor = remainder:
                                                              0012FF7B
  remainder = dividend % divisor;
                                                              0012FF7C
                                                              0012FF7D
                                             di vi dend
                                                         12
                                                              0012FF7E
                                                              0012FF7F
```

```
0012FF74
                                                              0012FF75
                                            remai nder
                                                              0012FF76
remainder = dividend % divisor;
                                                              0012FF77
while( remainder != 0 )
                                                              0012FF78
                                                              0012FF79
  di vi dend = di vi sor;
                                               di vi sor
                                                              0012FF7A
  divisor = remainder:
                                                              0012FF7B
  remainder = dividend % divisor;
                                                              0012FF7C
                                                              0012FF7D
                                             di vi dend
                                                         12
                                                              0012FF7E
                                                              0012FF7F
```

```
int main()
   int dividend, divisor, buffer;
   cin >> dividend >> divisor;
   if( divisor > dividend )
      buffer = divisor;
      di vi sor = di vi dend;
      di vi dend = buffer;
   int remainder = dividend % divisor;
   while( remainder != 0 )
      di vi dend = di vi sor;
      divisor = remainder;
      remainder = dividend % divisor;
   cout << divisor << endl << endl;
```